

METHODS, SYSTEMS AND COMPUTER PROGRAM PRODUCTS FOR
CONTROLLING DATA TRANSFER FOR DATA REPLICATION OR
BACKUP BASED ON SYSTEM AND/OR NETWORK RESOURCE
INFORMATION

Field of the Invention

The present invention relates to controlling data transfer on a data processing system and, more particularly, to controlling data replication or backup operations.

Background of the Invention

In some network environments, data is periodically transferred from one data processor to another data processor. Two examples of instances involving data transfer are data replication and data backup.

Data replication is the process of sharing changes between multiple copies of a single file or data set on multiple data processing systems. For example, two data processing systems can communicate through a network. Both systems on the network can have a copy of the same data. As time progresses, each data processing system updates its copy. After a period of time, the data may not be the same between data processing systems. Replication can be performed to

synchronize the data, and any updates to the data can be merged into one common data set.

Accordingly, data replication can allow users on various networks in a variety of locations to share the same information and files. Replication can be performed with replication software applications in a variety of processor relationships, such as client-server, server-server, and peer-peer replication.

Data replication is typically scheduled to occur at specified times. For example, replication can be scheduled to occur at a predetermined frequency, such as every two hours Monday through Friday and every four hours on weekends. However, the scheduling of data replication can disrupt system operations by using system resources. Occasionally, the scheduled data replication can occur at the same time that a data processing system is experiencing a heavy load. During such poorly timed replications, applications that may be running on the data processing system can be slowed. Response time for connecting users to the system may also be increased. In some instances, users may be unable to connect to the data processing system. The performance of applications sharing the same processor resources may also be adversely affected. Occasionally, the diversion of additional resources needed to perform data replication can result in an application failure as a result of low system resources.

Data transfers to back up data typically involve storing a copy of data separately from an original copy. For example, a document on a client data processing system can be saved periodically on a server processor. Many data backup systems are two tiered environments having a client machine with data that is backed up on a server machine. The client is scheduled

to send the data to the server at prescheduled times,
and the server manages the backed up data.

5 Data transfers to back up data typically occur at
regularly scheduled intervals. If the data backup
process is scheduled during a period of high processor
use, applications, connection times, and other system
performance measures may be adversely effected.

Summary of the Invention

According to embodiments of the present invention,
methods for controlling the periodic data transfer
between a first computer processor and a second
5 computer processor are provided. The first computer
processor and the second computer processor comprise a
network system. Data is transferred between the first
and second computer processor based on an impact of the
transfer on a dynamically determined measure of
10 performance.

In some embodiments of the present invention,
transferring data between the first and second computer
processor based on an impact of the transfer on a
measure of performance comprises monitoring a real time
15 availability of a system resource and transferring data
between the first and second computer processor based
on a comparison of the availability of a system
resource to a predetermined threshold. Data can be
transferred between the first and second computer
20 processor if a predefined maximum time between
transferring data has elapsed irrespective of the
availability of the network resource. The transfer of
data may be delayed until at least a predefined minimum
time has elapsed after a prior data transfer. A
25 resource availability request can be sent from the
first computer processor to the second computer
processor to determine resource availability.

Transferring data between the first and second computer processor can include replicating data and/or backing up data.

5 The system resource can be usage of the first and/or second computer processor, memory usage of the first and/or second computer processor, central processor unit (CPU) usage of the first and/or second computer processor, an available bandwidth on a network connection or any combination thereof.

10 While the invention has been described above primarily with respect to the method and data transfer control aspects of the invention, both systems and/or computer program products are also provided.

15 Brief Description of the Drawings

Figure 1 is a block diagram of a system according to embodiments of the present invention;

Figure 2 is a block diagram of a data processing system according to embodiments of the present invention;

20 **Figure 3** is a more detailed block diagram of data processing systems according to embodiments of the present invention; and

Figures 4-8 are flowcharts illustrating operations to control data transfer according to embodiments of the present invention.

Detailed Description

30 The present invention now will be described more fully hereinafter with reference to the accompanying drawings, in which embodiments of the invention are shown. This invention may, however, be embodied in many different forms and should not be construed as limited to the embodiments set forth herein; rather,

these embodiments are provided so that this disclosure will be thorough and complete, and will fully convey the scope of the invention to those skilled in the art.

As will be appreciated by those of skill in the art, the present invention may be embodied as methods, data processing systems, and/or computer program products. Accordingly, the present invention may take the form of an entirely hardware embodiment, an entirely software embodiment or an embodiment combining software and hardware aspects. Furthermore, the present invention may take the form of a computer program product on a computer-usable storage medium having computer-usable program code means embodied in the medium. Any suitable computer readable medium may be utilized including hard disks, CD-ROMs, optical storage devices, or magnetic storage devices.

Computer program code for carrying out operations of the present invention may be written in an object oriented programming language such as Java®, Smalltalk or C++. However, the computer program code for carrying out operations of the present invention may also be written in conventional procedural programming languages, such as the "C" programming language. The program code may execute entirely on the user's computer, partly on the user's computer, as a stand-alone software package, partly on the user's computer and partly on a remote computer or entirely on the remote computer. In the latter scenario, the remote computer may be connected to the user's computer through a local area network (LAN) or a wide area network (WAN), or the connection may be made to an external computer (for example, through the Internet using an Internet Service Provider).

The present invention is described below with reference to flowchart illustrations and/or block

diagrams of methods, apparatus (systems) and computer
program products according to particular embodiments of
the invention. It will be understood that each block
of the flowchart illustrations and/or block diagrams,
5 and combinations of blocks in the flowchart
illustrations and/or block diagrams, can be implemented
by computer program instructions. These computer
program instructions may be provided to a processor of
a general purpose computer, special purpose computer,
10 or other programmable data processing apparatus to
produce a machine, such that the instructions, which
execute via the processor of the computer or other
programmable data processing apparatus, create means
for implementing the functions specified in the
15 flowchart and/or block diagram block or blocks.

These computer program instructions may also be
stored in a computer-readable memory that can direct a
computer or other programmable data processing
apparatus to function in a particular manner, such that
20 the instructions stored in the computer-readable memory
produce an article of manufacture including instruction
means which implement the function specified in the
flowchart and/or block diagram block or blocks.

The computer program instructions may also be
25 loaded onto a computer or other programmable data
processing apparatus to cause a series of operational
steps to be performed on the computer or other
programmable apparatus to produce a computer
implemented process such that the instructions which
30 execute on the computer or other programmable apparatus
provide steps for implementing the functions specified
in the flowchart and/or block diagram block or blocks.

As is described in more detail below, the present
invention provides for the control of data transfer
35 operations, such as data replication and backup,

between first and second computer processors.

According to embodiments of the present invention, data can be transferred between computer processors based on an impact of the transfer on a dynamically determined measure of performance. In some embodiments of the invention, the impact of the transfer on the measure of performance can be determined based on the availability of a resource and a comparison that exceeds a threshold; however, other measures of performance can be used. For example, the availability of a resource and the availability threshold could also be a utilization and a utilization threshold or combinations thereof. Thus, availability includes monitoring utilization and/or the lack of utilization of a resource and does not refer to the present or absence of a resource in its totality.

The first computer processor and the second computer processor comprise a network system. An availability of a system resource is monitored. If the availability of the system resource is greater than a predetermined threshold, data is transferred between the first and second computer processor. Various embodiments of the present invention will now be described with reference to **Figures 1** through **8**.

Figure 1 illustrates a network environment in which embodiments of the present invention may be utilized. As will be appreciated by those of skill in the art, however, the operations of embodiments of the present invention may be carried out on a single processing system with or without access to a network such as an intranet or the Internet.

As seen in **Figure 1**, client data processing system(s) **12**, such as a personal computer, laptop computer, pervasive computing device such as a personal data assistant (PDA), smartphone, or other mobile

terminal, may communicate over a network 14 to another data processing system which may act as a server 16. The network 14 may be an intranet or the Internet or other networks known to those of skill in the art.

5 As is further illustrated in **Figure 1**, the server 16 can be in communication with data sources 30A, 30B, and 30C. The data sources 30A, 30B, and 30C can be computer servers, processing systems, or other networks that can send data to the client data processing system 12 over the network 14. A user can manipulate or view the data on the client 12.

10 Referring to **Figure 2**, exemplary embodiments of a data processing system 230 in accordance with embodiments of the present invention typically includes input device(s) 232 such as a keyboard or keypad, touch sensitive screen, light sensitive screen, or mouse, a display 234, and a memory 236 that communicate with a processor 238. The data processing system 230 may further include a speaker 244, and an I/O data port(s) 246 that also communicates with the processor 238. The I/O data port 246 can be used to transfer information between the data processing system 230 and another computer system or a network (e.g., the Internet). These components may be conventional components such as those used in many conventional data processing systems which may be configured to operate as described herein. Furthermore, as will be appreciated by those of skill in the art, the data processing system 230 may be configured as the client 12 or the server 16.

25 **Figure 3** is a block diagram of embodiments of data processing systems that illustrates systems, methods, and/or computer program products in accordance with the present invention. The processor 238

communicates with the memory 236 via an address/data bus 248. The processor 238 can be any commercially available or custom microprocessor. The memory 236 is representative of the overall hierarchy of memory devices containing the software and data used to implement the functionality of the data processing system 230. The memory 236 can include, but is not limited to, the following types of devices: cache, ROM, PROM, EPROM, EEPROM, flash memory, SRAM, and DRAM.

As shown in **Figure 3**, the memory 236 may include several categories of software and data used in the data processing system 230: the operating system 252; the application programs 254; the input/output (I/O) device drivers 258; and the data 256. As will be appreciated by those of skill in the art, the operating system 252 may be any operating system suitable for use with a data processing system, such as OS/2, AIX or System390 from International Business Machines Corporation, Armonk, NY, Windows95, Windows98, Windows2000, or WindowsXP from Microsoft Corporation, Redmond, WA, Unix or Linux. The I/O device drivers 258 typically include software routines accessed through the operating system 252 by the application program 254 to communicate with devices such as the input devices 232, the speaker 244, the I/O data port(s) 246, and certain components of the memory 236. The application programs 254 are illustrative of the programs that implement the various features of the data processing system 230 and preferably include at least one application which provides the data transfer control aspects of embodiments of the present invention. Finally, the data 256 represents the static and dynamic data used by the application programs 254, the operating system 252, the I/O device drivers 258, and

other software programs that may reside in the memory
236.

As is further seen in **Figure 3**, the application
programs 254 preferably include a data transfer control
5 module 260. The data transfer control module 260
preferably carries out operations as described herein
for controlling the transfer of data between
processors.

While the present invention is illustrated, for
10 example, with reference to a data transfer control
module 260, as will be appreciated by those of skill in
the art, the data transfer control module 260 may also
be incorporated into other components, such as the
operating system 252. Thus, the present invention
15 should not be construed as limited to the configuration
of **Figure 3** but is intended to encompass any
configuration capable of carrying out the operations
described herein.

Embodiments of the present invention will now be
20 described in more detail with reference to **Figure 4**,
which is a flowchart illustration of operations that
can be carried out by the data transfer control module
260. As seen in **Figure 4**, the data transfer control
module 260 may monitor an availability of a system
25 resource (Block 300). If the availability of the
resource is greater than a predetermined threshold
(Block 302), then data is transferred (Block 304).
Transferring data (Block 304) can include data
transfers to maintain multiple copies of a file in
30 separate memory locations on a system. Examples of
data transfer functions that can be performed include
data replication (to synchronize multiple copies of
files between processors) and backup (to store an
updated copy in a different location). If the

availability of the resource is less than a predetermined threshold, then no data is transferred (Block 306).

Accordingly, data transfers, such as replication and backup, are performed only if resources are available. The data transfer control module 260 in **Figure 3** can be self-configuring to determine an optimal time to perform data transfers, including data replication and backup.

A threshold amount of available resources that must be available for a data transfer to occur can be a preset value. Alternatively, the data transfer control module 260 can dynamically adjust the threshold amount of resource availability, for example, based on the duration of previous data transfers at a given resource availability. In some embodiments, the threshold can also be determined dynamically by an algorithm analyzing historical data. The system resource can include, for example, memory resource, central processor unit resources, timing resources, and/or bandwidth resources. The threshold can be a percentage of resource being used in real time, and the threshold can include timing information and/or other measurements of resource allocation and usage. Multiple resources may also be monitored and a composite of individual thresholds utilized. Moreover, the threshold can be dynamically adjusted over time. For example, the threshold could be decreased over time after each transfer so that the threshold is higher after a recent data transfer and lower as the time approaches the maximum time between transfers.

Moreover, data transfers can be controlled between two or more computer processors, such as the server 16 and one or more clients 12 as shown in **Figure 1**. For

example, the data transfer control module 260 (**Figure 3**) can reside on the server 16. The server 16 monitors resources, which can include resources that impact the server 16, the client 12, the network 14, or a combination thereof. Alternatively, the data transfer control module 260 can reside on the client 12 or the network 14.

For some data transfer operations such as data replication and backup, it may be desirable for data transfer to occur with a given regularity notwithstanding the availability of system resources. Referring to **Figure 5**, operations that can provide data transfers with a maximum time between data transfers are shown. The availability of a system resource is monitored (Block 500). If the availability of a resource is greater than a predetermined threshold (Block 502), then data is transferred (Block 506). However, if the availability of a resource is not greater than the predetermined amount, then the data transfer control module 260 determines if the maximum time between data transfers has been reached (Block 504). For example, the data transfer control module 260 can compare the time duration that has elapsed since the previous data transfer to a predetermined value. If the maximum time is reached, then data is transferred irrespective of resource availability (Block 506). If the maximum time has not been reached since the previous data transfer, then no transfer occurs (Block 508).

Performing some types of data transfer functions periodically, such as data replication and backup, may be necessary for system functionality. As illustrated in **Figure 5**, the data transfer generally occurs based on the determination of resource availability (Block

502). However, if a maximum time has been reached, then the data can be transferred regardless of resource availability. Accordingly, data transfer occurs at least as often as the maximum time interval.

5 In some applications, the data transfer control module 260 can provide data transfers that have a minimum amount of time between transfers. There may be no advantage to performing data replication more frequently than a given amount, such as thirty or sixty
10 minutes. Moreover, data replication performed too frequently may use system resources that could be available for other applications. Operations to provide a minimum time between transfers are shown in **Figure 6**.

15 As shown, if a minimum time between transfers has not elapsed (Block 600), then no data transfer occurs (Block 602). However, if the minimum time between transfers has elapsed, then the data transfer control module 260 monitors an availability of a system
20 resource (Block 604). If the resource availability is greater than a predetermined threshold, then data transfer operations are initiated (Block 610). If the resource availability is not greater than the predetermined threshold, then data is not transferred
25 (block 608).

 As shown in **Figure 7**, the data transfer control module 260 can provide both a minimum and a maximum time in between data transfer operations. Accordingly, data transfer can occur based on resource availability,
30 but also between a minimum and maximum time regardless of resource availability. If the minimum time between transfers has not elapsed (Block 704), then there is no data transfer (Block 702). After the minimum time between transfers has elapsed, then the data transfer

control module 260 monitors the availability of a resource (Block 700). If the resource availability is greater than a predetermined threshold (Block 706), then data is transferred (Block 712). If the resource
5 availability is not greater than the predetermined threshold (Block 706) and the maximum time between transfers has not elapsed (Block 708), then there is no data transfer. However, if the maximum time between transfers has elapsed (Block 708), then data is
10 transferred (Block 712).

Communications between two or more processors or resources in the system can be used to determine the availability of a resource. As shown in **Figure 8**, if a minimum time between transfers has elapsed (Block 800),
15 then one processor or resource, such as a server 16 as shown in **Figure 1**, can monitor an availability of a server resource, such as server memory, processor usage, and the like (Block 804). If the server resource availability is greater than a predetermined amount (Block 806), then a negotiable transfer request
20 is sent, for example, from the server 16 to the client 12 (Block 808). The client 12 may send a response to the server 16 indicating whether the client has resources available for data transfer operations. If
25 the response is positive (Block 812), then data is transferred (Block 816). If the response is negative, then data is not transferred (Block 814).

Accordingly, if the resource availability is greater than a threshold with respect to one processor,
30 the processor can send a request to a second processor. The second processor can respond with either a positive or negative response, indicating that its resource availability is also greater than a selected threshold

amount. The threshold amounts with respect to the various processors can be different or the same, and the resource availability queries can involve the same type of resource (memory usage, processor usage,
5 bandwidth, etc.) or different types of resources or combinations of resources on the system.

As further shown in **Figure 8**, additional operations can include controlling data transfers according to minimum and maximum times between
10 transfers. If the minimum time between transfers has not been met (Block 800), then data transfers do not occur. (Block 802). If a maximum time between transfers has elapsed (Block 810), then data transfers occur (Block 816).

15 While the present invention has been described with reference to a client-server architecture, as will be appreciated by those of skill in the art, the present invention may be provided as a stand-alone application.

20 In the drawings and specification, there have been disclosed typical preferred embodiments of the invention and, although specific terms are employed, they are used in a generic and descriptive sense only and not for purposes of limitation, the scope of the
25 invention being set forth in the following claims.